

Question 1

Input [7:0] A, B;

Input sel;

Output [7:0] F;

實作方式：

(1) 邏輯

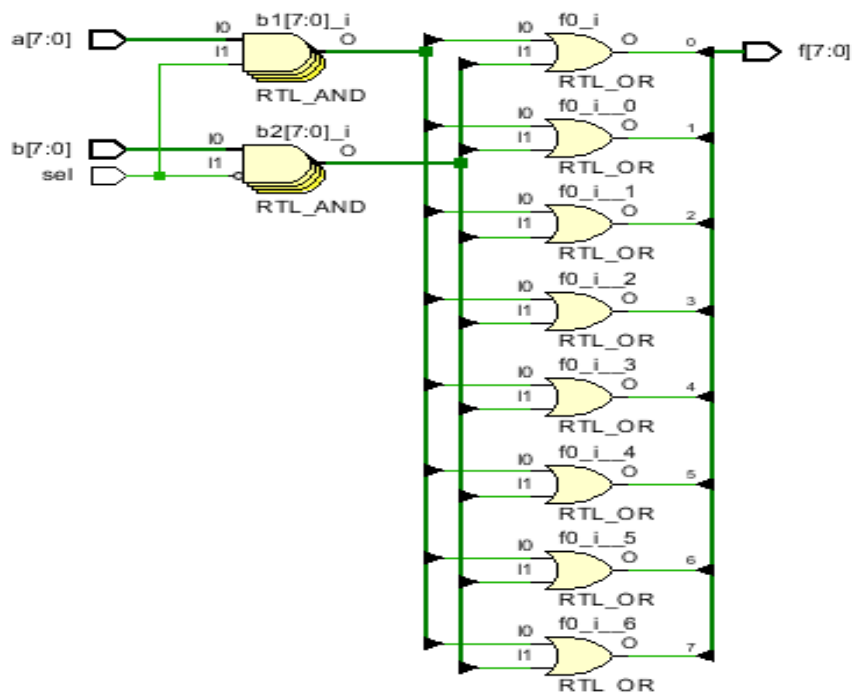
將每一位分別獨立出來，若 sel 訊號為 1 則該位數選取 A，若 sel 訊號為 0 則該位數選取 B。寫為布林代數式即為：

$$F[i] = A[i] \text{ sel} + B[i] \text{ sel}' \quad , (\text{where } 0 \leq i \leq 7)$$

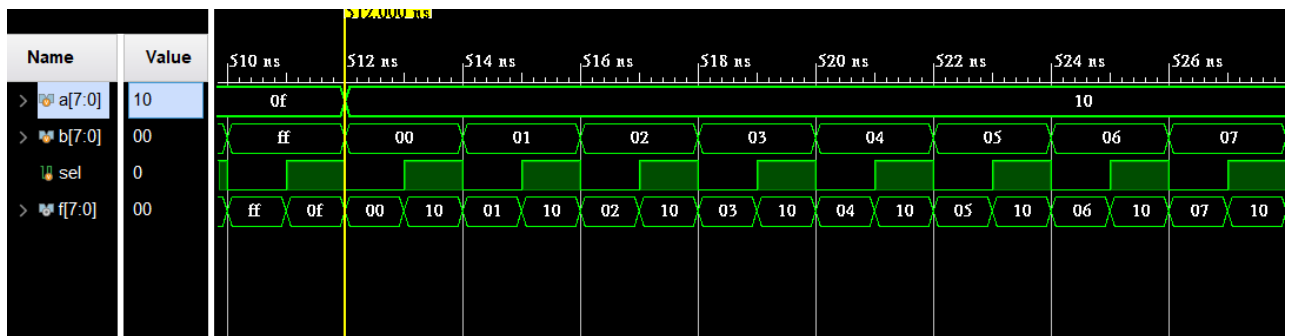
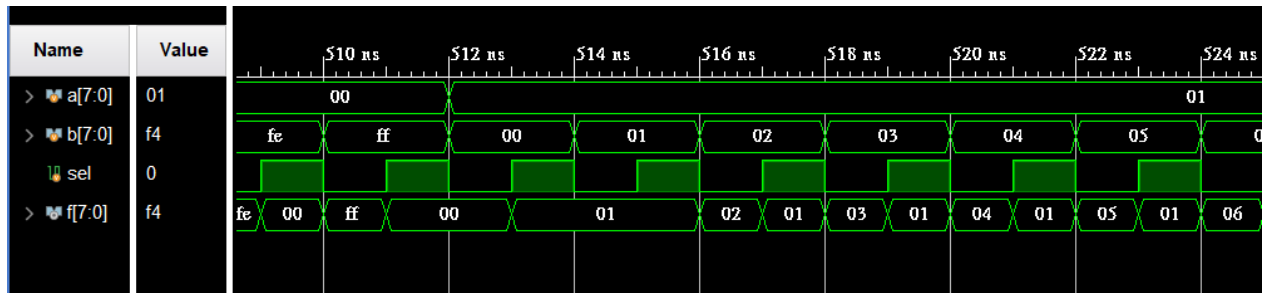
(2) 轉換為線路圖

先用一個 and gate 分別將該位數的 a, sel 串接，另一個 and gate 將該位數的 B, sel' 串接，最後用 or gate 將前兩項的結果輸出，作為該位數的輸出

以下為線路圖:



(3) 用 verilog 模擬



模擬後與預期結果相同，若 $sel = 1$ ，輸出結果為 8 - bits A，若 $sel = 0$ ，輸出結果為 8 - bits B。

Question 2

$Din[3:0]$ — 4x16 decoder — $Dout[15:0]$

Input $Din[3:0]$	Output $Dout[15:0]$	Input $Din[3:0]$	Output $Dout[15:0]$
1111	0000_0000_0000_0001	0111	1000_0000_0000_0000
1110	0000_0000_0000_0010	0110	0100_0000_0000_0000
1101	0000_0000_0000_0100	0101	0010_0000_0000_0000
1100	0000_0000_0000_1000	0100	0001_0000_0000_0000
1011	0000_0000_0001_0000	0011	0000_1000_0000_0000
1010	0000_0000_0010_0000	0010	0000_0100_0000_0000
1001	0000_0000_0100_0000	0001	0000_0010_0000_0000
1000	0000_0000_1000_0000	0000	0000_0001_0000_0000

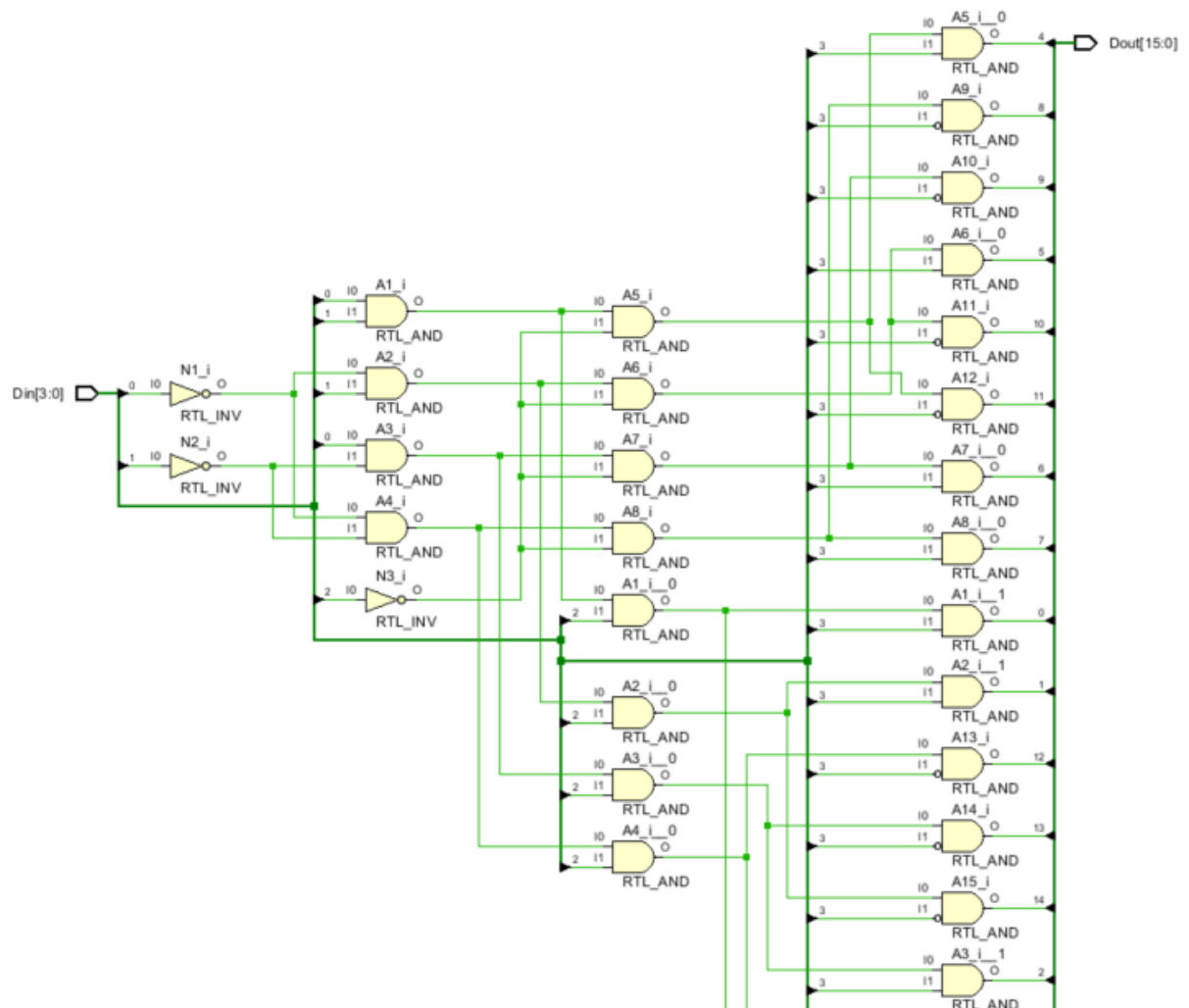
由以上的題幹可得到真值表。例如：只有在 Din 為 1111 時， $Dout[0] = 1$ ，因此可得到相對應的 logic equation： $Dout[0] = Din[3] \& Din[2] \& Din[1] \& Din[0]$ 。

同理，

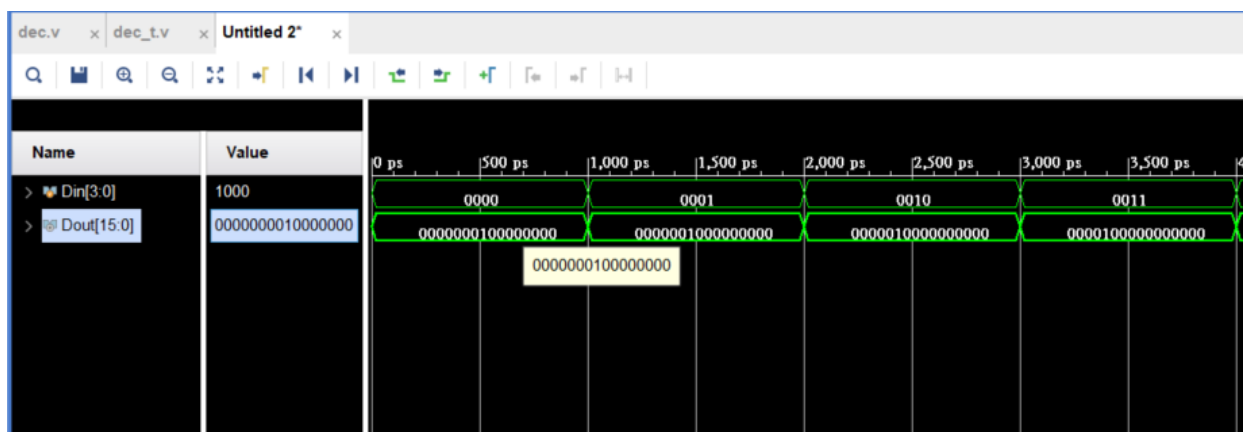
$Din = 1110$ 時， $Dout[1] = 1$ 因此 $Dout[1] = Din[3] \& Din[2] \& Din[1] \& (\sim Din[0])$ ，

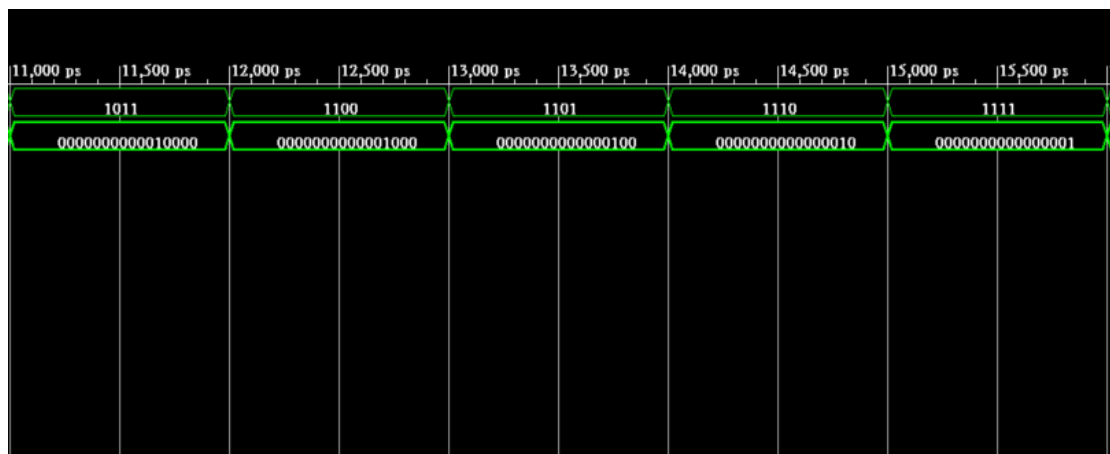
Din = 1101 時，Dout[2] = 1 因此 $Dout[2] = Din[3] \& Din[2] \& (\sim Din[1]) \& Din[0]$ ，
Din = 1100 時，Dout[3] = 1 因此 $Dout[3] = Din[3] \& Din[2] \& (\sim Din[1]) \& (\sim Din[0])$ ，
Din = 1011 時，Dout[4] = 1 因此 $Dout[4] = Din[3] \& (\sim Din[2]) \& Din[1] \& Din[0]$
等等……。

因此，由以上的 logic equation 可以畫出以下對應的 logic diagram:

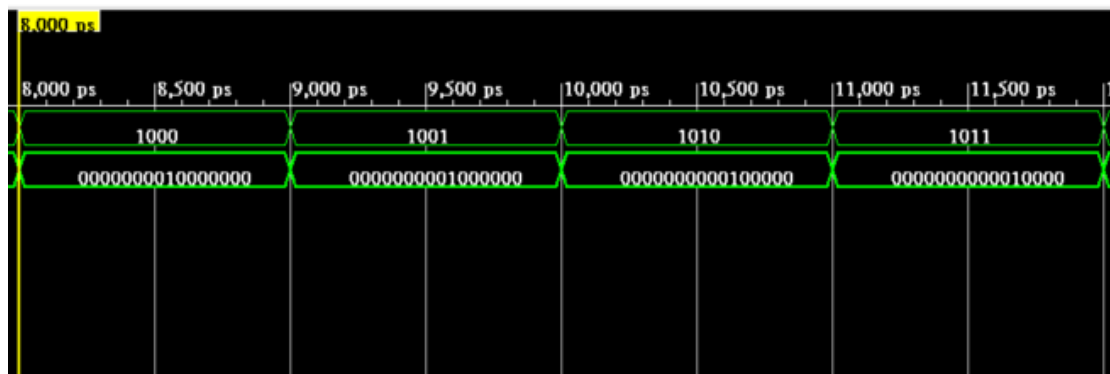


由 Vivado 可模擬出以下的波形圖：





如上圖所示，當 Din = 1111 時，Dout = 0000000000000001(Dout[0] = 1),
 Din = 1110 時，Dout = 0000000000000010(Dout[1] = 1),
 Din = 1100 時，Dout = 0000000000000100(Dout[3] = 1),
 ...



Din = 1000 時，Dout = 0000000010000000(Dout[7] = 1),
 ...

與題幹所給的 truth table 完全吻合。

此題的檢測方式為，在 testbench 裡中，預設 Din = 0000，在每一次的迴圈中都加上 4' b0001，總共執行了 2 的 4 次方次(會從 4' b0000 加到 4' b1111)，如此一來就能檢測到所有的可能的測資。

Question 3

Input [2:0]A, B;

Output a_lt_b, a_gt_b, a_eq_b;

實作方式：

(1) 分別解決 AB 大小的三種情況

Case $A > B$:

即為 A[2]大於 B[2]，或 A[2]等於 B[2]且 A[1] > B[1]，或 A[2]等於 B[2]且 A[1]等於 B[1]且 A[0] > B[0]。

寫為布林代數即為 $A[2]B[2]' + (A[2]B[2] + A[2]'B[2])A[1]B[1]' + (A[2]B[2] + A[2]'B[2]')(A[1]B[1] + A[1]'B[1]')A[0]'B[0]$

Case $A < B$:

同理於 $A > B$ ，布林代數式為 $A[2]'B[2] + (A[2]B[2] + A[2]'B[2])A[1]'B[1] + (A[2]B[2] + A[2]'B[2]')(A[1]B[1] + A[1]'B[1]')A[0]'B[0]$

Case $A = B$:

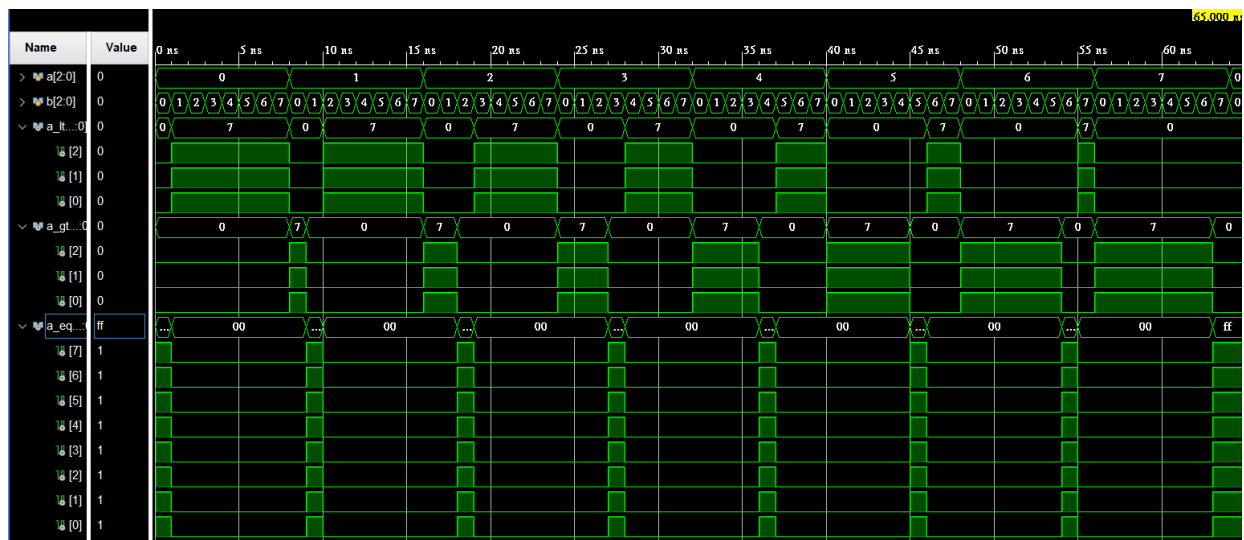
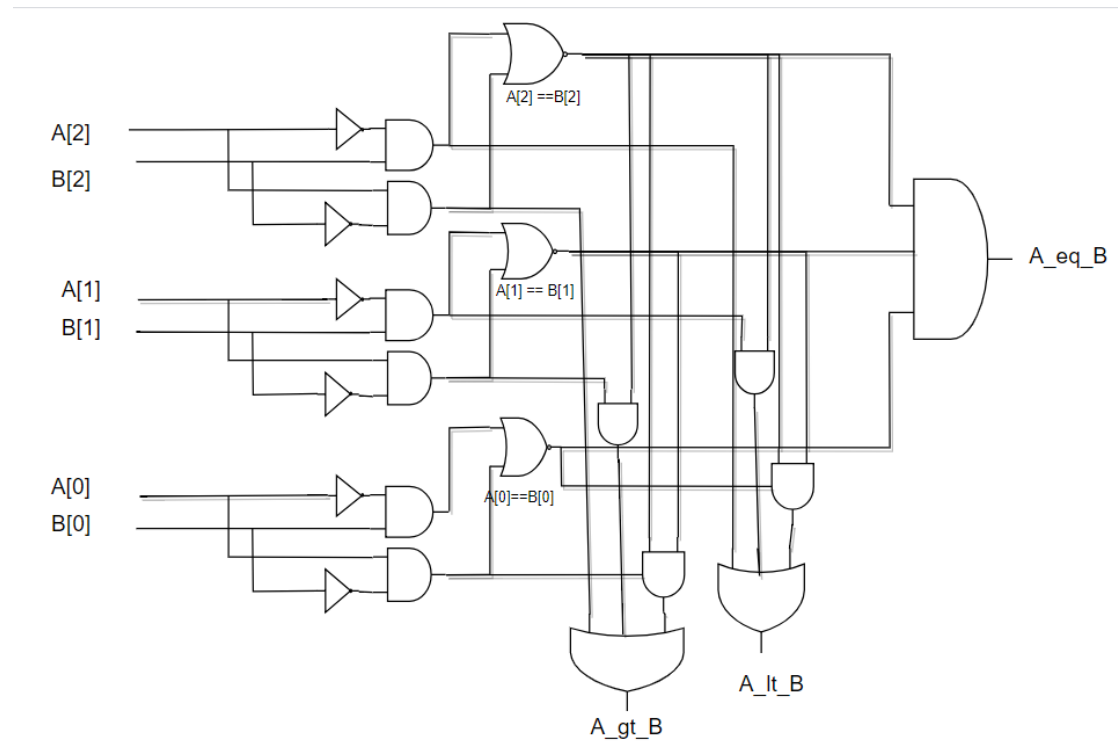
為三個位數接相等的情況。布林代數為 $A[2]B[2] + A[2]'B[2]' + A[1]B[1] + A[1]'B[1]' + A[0]B[0] + A[0]'B[0]'$

(2) 將上述想法轉換為線路圖

首先將每一位數的 A 與 B 用 AND gate 串接為 $A'B$ 和 AB' ，其分別代表該位數 A 小於 B 及 A 大於 B 的情況，然後將 $A'B$ 及 AB' 的線拉出接到 nor gate 上，nor gate 的意義即為該位數 $A = B$ 的情況。最後依照上述的布林代數式接到不同邏輯閘然後輸出到 3 個 output 上。

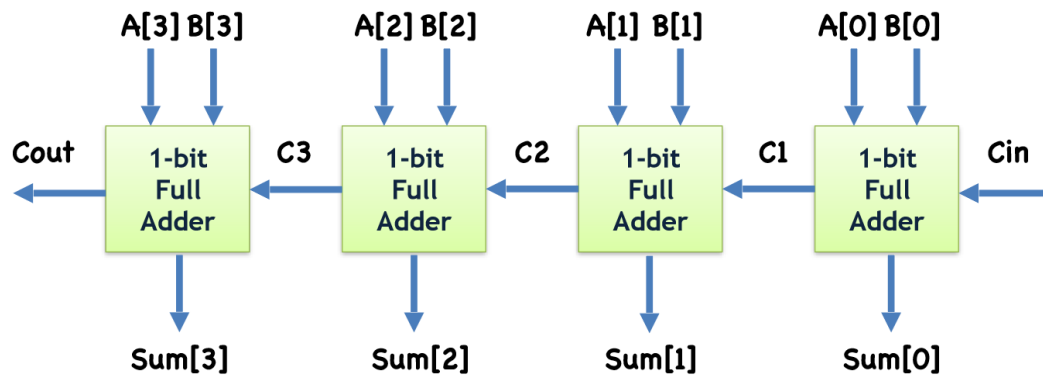
以下為線路圖

(3) 將以上電路圖轉換為 verilog 後模擬



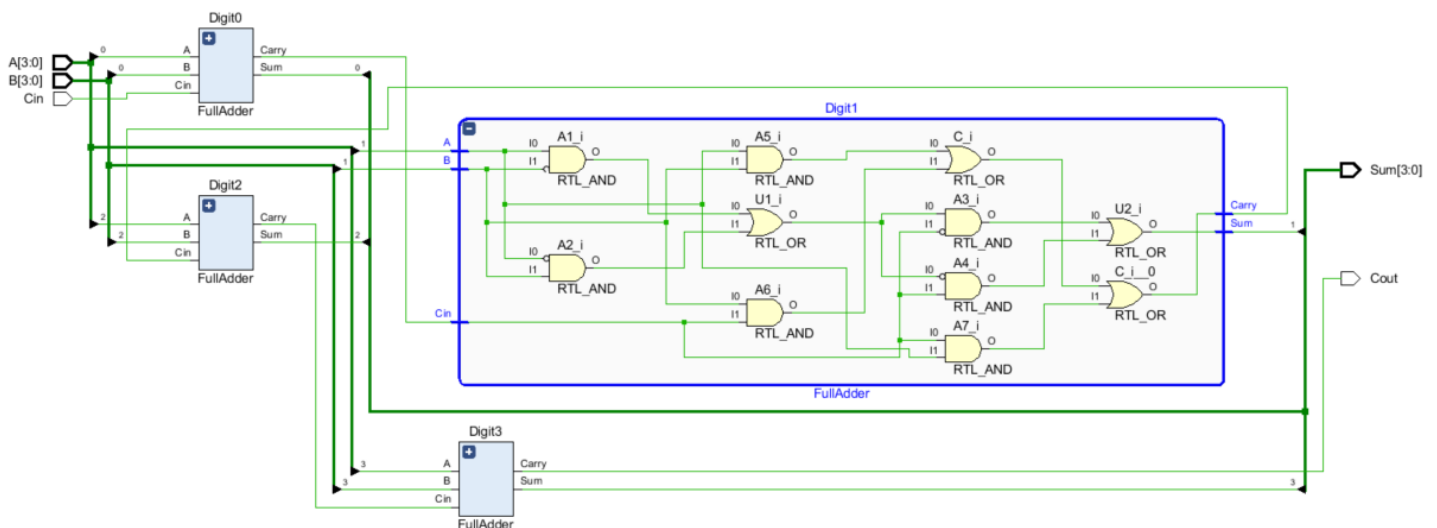
模擬後與預期得到相同的結果， $A > B$ 時 a_{gt_b} output 為一，其 他情況為零， $A < B$ 時 a_{gt_b} output 為一，其他情況為零， $A = B$ 時 a_{eq_b} output 為一，其他情況為零。

Question 4

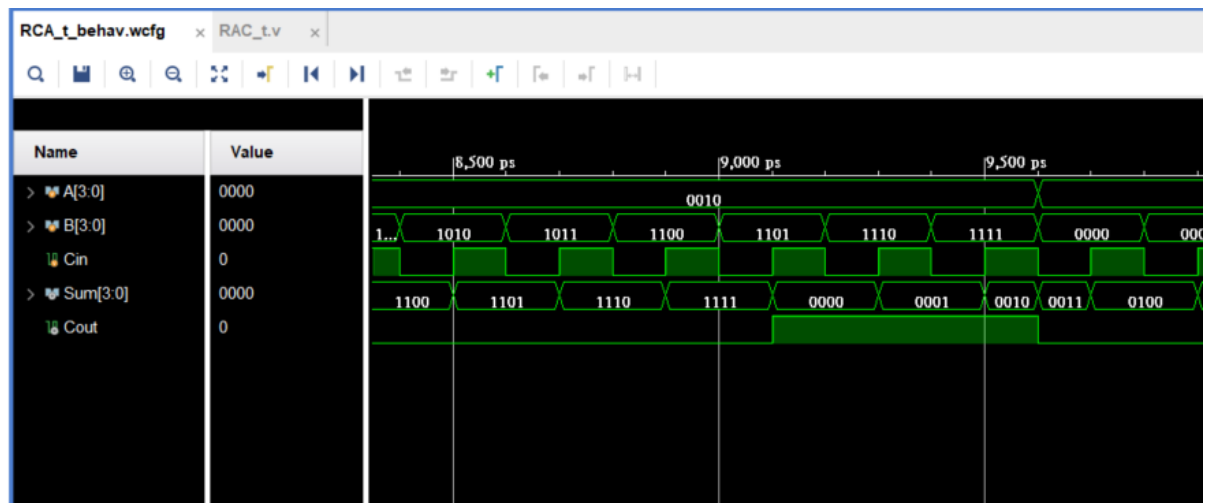


想法：在 Lab1 Basic Question 中已經得知 1-bit full adder 的架構，其 logic equation 為： $Sum = A \oplus B \oplus C_{in}$ ， $Carry = A * B + B * C_{in} + C_{in} * A$ (Carry 為進位數，以上圖來看可能是 C1, C2, C3 或 Cout)。因此，欲製作 4-bit ripple carry adder，需要 4 個 full adder 將每個位數所對應到的 bit 彼此相加，然後將前一位需要進位的數字加到後一位，如同上圖所示。

因此，參照以上的邏輯可以畫出下方的 logic diagram:

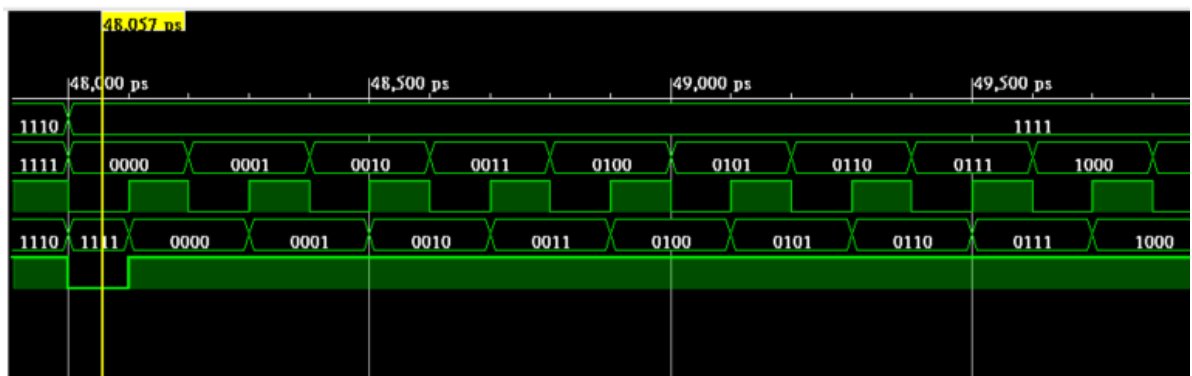


由 Vivado 可模擬出以下的波形圖：



如上圖所示，當 $A = 0010$, $B = 1011$, $Cin = 0$ 時， $Sum = 1101$, $Cout = 0$;
 $0010 + 1011 + 0 = 1101$ 答案正確。

當 $A = 0010$, $B = 1011$, $Cin = 1$ 時， $Sum = 1110$, $Cout = 0$;
 $0010 + 1011 + 1 = 1110$ 答案正確。



B, Cin 結合起來，視為一個 9-bit 的數字，將在每一次的迴圈中都加上
9' b000000001，總共執行了 2 的 9 次方次(會從 9' b000000000 加到
9' b111111111)，如此一來就能檢測到所有的可能的測資。

心得

距離我上次學邏輯設計已經是大一上的事情，面對大二上修邏輯設計實驗，一開始真的覺得非常的頭痛，不僅僅是這學期的主軸 verilog 不會打、基本的 Mux、Decoder、Full Adder 都忘的一乾二淨，第一次到資電館的 Lab1 可以說是糟透了。但是還好在短短的一個禮拜內，我加緊腳步複習，也幸好有許多大一下修過邏輯設計實驗的人以及我的組員給予我協助，讓我能順利完成這一次的 Lab。回憶起以前學邏輯設計，當時因為沒什麼在打 verilog，以至於對所教的東西，概念上總是顯得特別模糊，但是這次的 Lab 讓完全了解 Mux、Decoder、Comparator 以及 Ripple Carry Adder，整體來說學到很多東西(雖然 Lab1 很簡單☹) 這一次的突破，讓我有信心面對下一次的 Lab。(107060011 涂皓鈞)

這次實驗時間花最久的就是畫圖了，因為對軟體實在是有夠不熟，所以畫那些邏輯閘花了很多時間，希望以後熟一點可以畫得越來越快。這次的邏輯設計實驗也是充滿挑戰，因為我是大一上時修邏輯設計的，所以等到二上修邏輯設計實驗時都已經有點生疏，不過還好藉由第一次的 Lab 喚起我以前的一點記憶及打 verilog 的感覺。這次實驗的每一題都有打出來，得到了一些成就感，希望以後的邏設實驗也可以這麼順利。(107060015 陳弘輕)

小組分工

涂皓鈞 - 負責 Question2、4 的 verilog 以及對應題號的結報內容 + 自己的心得
陳弘輕 - 負責 Question1、3 的 verilog 以及對應題號的結報內容 + 自己的心得